

Datos experimentales:

La unidad de distancia es el diámetro de la partícula y la unidad de tiempo el fotograma. Solo se incluyen trayectorias dentro de una región de interés (ROI) circular con centro en ( $x = 12$ ,  $y = 7$ ) y radio  $R = 6$ .

Algunos otros datos de interés :

diámetro de la partícula = 56 píxeles (es una medida provisional)

diámetro de la partícula = 40 mm

duración de un fotograma = 1/400 s

masa ppp=2.8 gramos

### Funciones y parámetros a ejecutar/introducir

Ojo: `carpetaDatos` contiene la dirección de la carpeta donde están los datos procesados mediante el notebook “Import Export datos....”.

Nombre de la carpeta donde está el fichero con los datos experimentales procesados mediante el notebook “Import Export datos....”.

```
In[1]:= carpetaDatos = "F:\\Dropbox\\Documentos\\DIFU\\PingPong\\Datos Experimentos";
```

La variable `carpetaDatos` contiene la dirección del directorio donde están guardados los datos experimentales en la forma de un fichero para la serie de datos de cada partícula. Estos ficheros se generaron mediante el notebook “Import Export datos....”.

El formato de los datos es {posición x, posición y, frame (~tiempo), número de partícula}

Por tanto la columna de la posición x es la 1, `colPosX=1`

Por tanto la columna de la posición y es la 2, `colPosY=2`

```
In[2]:= colPosX = 1; colPosY = 2;
```

```
In[3]:= duracionFotograma = 1 / 400; (* en segundos*)
```

`listaParticulasconNumerodeDatos` es una tabla que contiene el “nombre” (número) de la partícula/ppp junto con la longitud de la serie de datos que se han medido. Su formato es {Nombre partícula, longitud lista de datos experimentales tomados}

Se generó con el notebook Import Export datos ...

Seleccionamos las partículas cuyas series de datos tengan una longitud mayor o igual que `ncorte`. Esto lo hace la función `listaParticulasconNumerodeDatosMayorolgualque`

```
In[5]:= listaParticulasconNumerodeDatosMayorolgualque[ncorte_] :=  
Select[listaParticulasconNumerodeDatos, #[[2]] >= ncorte &]
```

Ejemplo

```
In[6]:= listaParticulasconNumerodeDatosMayorolgualque[10 000] // Length
```

```
Out[6]= 32
```

Seleccionamos las partículas cuyas series de datos tengan una longitud

mayor o igual que ncorte y damos el resultado ordenado empezando por las series MÁS LARGAS. Esto lo hace la función listaORDENADASdeParticulasconNumerodeDatosMayorolqualque

```
In[7]:= listaORDENADASdeParticulasconNumerodeDatosMayorolqualque[ncorte_] :=
  Sort[Table[{listaParticulasconNumerodeDatosMayoroIgualque[ncorte][[m, 1]],
    listaParticulasconNumerodeDatosMayoroIgualque[ncorte][[m, 2]]},
   {m, 1, Length[listaParticulasconNumerodeDatosMayoroIgualque[ncorte]]}],
  #1[[2]] > #2[[2]] &]
```

datosdeParticula[nombreParticula\_]: Función que carga el archivo de datos de la partícula nombreParticula y asigna estos valores (en forma de tabla con elementos {x,y,frame, nombreParticula}) a datosdeParticula

```
In[8]:= datosdeParticula[nombreParticula_] := datosdeParticula[nombreParticula] =
  Module[{archivo, datosdeParticulaBrutos},
  archivo = carpetaDatos <>
    "\\\\'\\">> ToString[nombreParticula]<> ".dat";
  Import[archivo]
  ];
```

PosicionLista[nombreParticula\_, colPos\_] :::: Da la posición X si colPos=colPosX (=colPosX=1) o la posición Y si colPos(=colPosY=2 si está en la columna 2) de la partícula

```
In[9]:= PosicionLista[nombreParticula_, "X"] := PosicionLista[nombreParticula, "X"] =
  Table[datosdeParticula[nombreParticula][[n, colPosX]],
  {n, 1, Length[datosdeParticula[nombreParticula]]}];
```

```
In[10]:= PosicionLista[nombreParticula_, "Y"] := PosicionLista[nombreParticula, "Y"] =
  Table[datosdeParticula[nombreParticula][[n, colPosY]],
  {n, 1, Length[datosdeParticula[nombreParticula]]}];
```

PosicionLista[nombreParticula\_, {colPosX\_, colPosY\_}] :::: Da la posición {X,Y} si colPos=colPosX (=colPosX=1) o la posición Y si colPos(=colPosY=2 si está en la columna 2) de la partícula

```
In[11]:= PosicionLista[nombreParticula_, {"X", "Y"}] :=
  PosicionLista[nombreParticula, {"X", "Y"}] =
  Table[{datosdeParticula[nombreParticula][[n, colPosX]],
  datosdeParticula[nombreParticula][[n, colPosY]]},
  {n, 1, Length[datosdeParticula[nombreParticula]]}];
```

DeltaPosicionLista[nombreParticula,deltan] es la lista de cambios en la posición y de la ppp "nombreParticula" tras deltan frames, es decir

{.....,y(frame+deltan)-y(frame),,y(frame+2deltan)-y(frame+deltan),.....}

**POR TANTO , LA SERIE SE \*DIEZMA\*, ES DECIR, EL NÚMERO DE ELEMENTOS/DATOS DE LA SERIE SE REDUCE EN FACTOR deltan**

```
In[12]:= datosdeParticula[518]
```

```
Out[12]= {{11.3551, 12.9642, 19 992, 518}, {11.3527, 12.962, 19 993, 518},  

{11.351, 12.9576, 19 994, 518}, {11.349, 12.955, 19 995, 518},  

{11.3466, 12.9526, 19 996, 518}, {11.3456, 12.9487, 19 997, 518},  

{11.3444, 12.9472, 19 998, 518}, {11.3412, 12.9447, 19 999, 518},  

{11.3391, 12.942, 20 000, 518}, {11.3371, 12.9397, 20 001, 518}}
```

```
In[13]:= Clear[DeltaPosicionLista]
```

```
In[14]:= DeltaPosicionLista[nombreParticula_, deltan_, eje_] :=  

DeltaPosicionLista[nombreParticula, deltan, eje] =  

Table[PosicionLista[nombreParticula, eje][[n + deltan]] -  

PosicionLista[nombreParticula, eje][[n]],  

{n, 1, Length[PosicionLista[nombreParticula, eje]] - deltan, deltan}]
```

```
In[15]:= DeltaPosicionLista[nombreParticula_, deltan_, {"X", "Y"}] :=  

DeltaPosicionLista[nombreParticula, deltan, {"X", "Y"}] =  

Transpose[{DeltaPosicionLista[nombreParticula, deltan, "X"],  

DeltaPosicionLista[nombreParticula, deltan, "Y"]}]]
```

Una única serie en la que se unen las listas que contienen los saltos espaciales (correspondientes a un tiempo de viaje igual a deltan ) en la dirección X o Y para series con una longitud mínima dada

```
In[16]:= serieTotaldeDeltaPos[longitudMinimadelasSeries_, deltan_, eje_] :=  

serieTotaldeDeltaPos[longitudMinimadelasSeries, deltan, eje] =  

Module[{listaParticulasSeleccionadas},  

listaParticulasSeleccionadas =  

listaParticulasconNumerodeDatosMayoroIgualque[  

longitudMinimadelasSeries];  

Flatten[Table[DeltaPosicionLista[listaParticulasSeleccionadas[[m, 1]],  

deltan, eje], {m, 1, Length[listaParticulasSeleccionadas]}]]]  

]
```

DeltaPosicionListaCOMPLETA[nombreParticula,deltan,"X"] es la lista de cambios en la posición x de la ppp "nombreParticula" tras deltan frames, es decir

{.....,y(frame+deltan)-y(frame),,y(frame+2deltan)-y(frame+deltan),.....}

pero \*sin DIEZMAR\* , EL NÚMERO DE ELEMENTOS/DATOS DE LA SERIE SOLO SE REDUCE EN deltan elementos

```
In[17]:= DeltaPosicionListaCOMPLETA[nombreParticula_, deltan_, eje_] :=  

DeltaPosicionListaCOMPLETA[nombreParticula, deltan, eje] =  

Differences[PosicionLista[nombreParticula, eje], 1, deltan]
```

```
In[18]:= serieTotaldeDeltaPosCOMPLETA[longitudMinimadelasSeries_, deltan_, eje_] :=
serieTotaldeDeltaPosCOMPLETA[longitudMinimadelasSeries, deltan, eje] =
Module[{listaParticulasSeleccionadas},
listaParticulasSeleccionadas =
listaParticulasconNumerodeDatosMayoroIgualque[
longitudMinimadelasSeries];
Flatten[Table[DeltaPosicionListaCOMPLETA[listaParticulasSeleccionadas[[m, 1]], deltan, eje], {m, 1, Length[listaParticulasSeleccionadas]}]]
]
```

### Ejemplo

```
In[19]:= {PosicionLista[518, "X"],
PosicionLista[518, "Y"], DeltaPosicionLista[518, 2, "X"],
DeltaPosicionListaCOMPLETA[518, 2, "X"]} // TableForm
```

Out[19]/TableForm=

	11.3551	11.3527	11.351	11.349	11.3466	11.3456
12.9642	12.962	12.9576	12.955	12.9526	12.9487	
-0.00409669	-0.00438938	-0.00225508	-0.00531837			
-0.00409669	-0.00368051	-0.00438938	-0.00339175	-0.00225508	-0.0041	

timeAverageMSD[ListaTemporaldeAlgo, ventanaTemporal] evalúa la serie/lista de diferencias de Algo entre posiciones separadas por ventanaTemporal (ventana temporal de tamaño ventanaTemporal), las eleva al cuadrado y halla el valor medio (TimeAverageMSD). El resultado es un número que depende del valor de ventanaTemporal. Esto es lo que se conoce como promedio de una serie temporal recorriendo los tiempos

```
In[20]:= timeAverageMSDySTD[data_, dn_] := Module[{diff, y2m, y2s},
diff = (Differences[data, 1, dn]^2);
{y2m, y2s} = {Mean[#], StandardDeviation[#]} &[diff]];
```

```
In[21]:= timeAverageMSD[data_, dn_] := Module[{diff},
diff = (Differences[data, 1, dn]^2);
Mean[diff]];
```

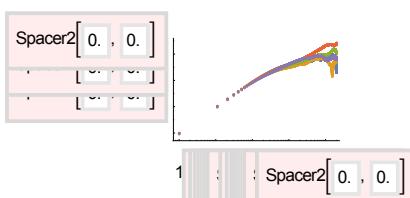
```
In[22]:= timeAverageDisplacement[data_, dn_] := Module[{diff},
diff = Differences[data, 1, dn];
Mean[diff]];
```

seriesdeTAMSD evalúa una tabla de tablas, cada una de esta subtablas formada de elementos con formato {tamañoVentanaTemporal, timeAveragedeIMSDdelaSerieConEsaVentanaTemporal}. Cada subtabla corresponde a una partícula que tiene una serie con longitud mayor o igual que longitudMinimadelasSeries. El tamaño de la ventana temporal crece en intervalos de deltanIncre

```
In[23]:= seriesdeTAMSD[longitudMinimadelasSeries_, deltanIncre_, eje_] :=
seriesdeTAMSD[longitudMinimadelasSeries, deltanIncre, eje] =
Module[{listaParticulasSeleccionadas, dnfinal},
 listaParticulasSeleccionadas =
 listaParticulasconNumerodeDatosMayoroIgualque[
 longitudMinimadelasSeries];
Table[Print["Generando TAMSD de la serie... ",
 listaParticulasSeleccionadas[[m, 1]]];
 Table[{dn, timeAverageMSD[PosicionLista[listaParticulasSeleccionadas[[m, 1]], eje], dn]}, {dn, 1, longitudMinimadelasSeries - dltanIncre, dltanIncre}], {m, 1, Length[listaParticulasSeleccionadas]}]
]
```

```
In[24]:= seriesdeTAMDisplacement[longitudMinimadelasSeries_, dltanIncre_, eje_] :=
seriesdeTAMDisplacement[longitudMinimadelasSeries, dltanIncre, eje] =
Module[{listaParticulasSeleccionadas, dnfinal},
 listaParticulasSeleccionadas =
 listaParticulasconNumerodeDatosMayoroIgualque[
 longitudMinimadelasSeries];
Table[Print["Generando TAD (time average displacement) de la serie... ",
 listaParticulasSeleccionadas[[m, 1]]];
 Table[{dn, timeAverageDisplacement[PosicionLista[
 listaParticulasSeleccionadas[[m, 1]], eje], dn]},
 {dn, 1, longitudMinimadelasSeries - dltanIncre, dltanIncre}],
 {m, 1, Length[listaParticulasSeleccionadas]}]
]
```

(\* Mostramos lo que es seriesdeTAMSD \*)  
ListLogLogPlot[seriesdeTAMSD[20 000, 10, "X"] ] -->



VLista[nombreParticula\_, lag\_, deltaDif\_, eje] da una lista con la velocidad “eje” de la partícula nombreParticula evaluada mediante diferencias finita central con  $\Delta=\text{deltaDif}$ , evaluándose en las posiciones 1,1+lag,1+2lag,1+3lag,...El valor de eje es “X” o “Y”

```
In[25]:= VLista[nombreParticula_, lag_, deltaDif_, eje_] := Prepend[
Table[(PosicionLista[nombreParticula, eje][[n + deltaDif]] -
 PosicionLista[nombreParticula, eje][[n - deltaDif]])/
(2 * deltaDif * duracionFotograma), {n, lag + 1,
Length[datosdeParticula[nombreParticula]] - lag, lag}],
(PosicionLista[nombreParticula, eje][[1 + deltaDif]] -
 PosicionLista[nombreParticula, eje][[1]]) /
(deltaDif * duracionFotograma)];
```

Ejemplos

```
In[26]:= VLista[508, 2, 2, "X"]
Out[26]= {-1.25781, -1.14915, -1.17416, -1.36086}
```

```
In[27]:= VLista[508, 1, 1, "X"]
Out[27]= {-1.60654, -1.25781, -1.13682, -1.0405,
-1.12477, -1.30782, -1.3849, -1.41389, -1.10535}
```

```
In[28]:= VLista[nombreParticula_, lag_, deltaDif_, eje_, {frameIni_, frameFin_}] :=
If[frameFin < Length[datosdeParticula[nombreParticula]],
Table[(PosicionLista[nombreParticula, eje][[n + deltaDif]]] -
PosicionLista[nombreParticula, eje][[n - deltaDif]])/
(2 * deltaDif * duracionFotograma), {n, frameIni + lag, frameFin - lag, lag}],
Table[(PosicionLista[nombreParticula, eje][[n + deltaDif]]] -
PosicionLista[nombreParticula, eje][[n - deltaDif]])/
(2 * deltaDif * duracionFotograma), {n, frameIni + lag,
Length[datosdeParticula[nombreParticula]] - lag, lag}]]
```

```
In[29]:= serieTotaldeV[longitudMinimadelasSeries_, lag_, deltaDif_, eje_] :=
serieTotaldev[longitudMinimadelasSeries, lag, deltaDif, eje] =
Module[{listaParticulasSeleccionadas},
listaParticulasSeleccionadas =
listaParticulasconNumerodeDatosMayoroIgualque[
longitudMinimadelasSeries];
Flatten[Table[VLista[listaParticulasSeleccionadas[[m, 1]], lag,
deltaDif, eje], {m, 1, Length[listaParticulasSeleccionadas]}]]
]
```

RefLink [ CorrelationFunction , paclet : ref / CorrelationFunction ] [

$\{x_1, \dots, x_n\}, h$  ] is equivalent to  $\sum_{i=1}^{n-h} (x_i - \hat{\mu}) (x_{i+h} - \hat{\mu}) / \sum_{i=1}^n (x_i - \hat{\mu})^2$  with  $\hat{\mu} =$

RefLink [ Mean , paclet : ref / Mean ] [  $\{x_1, \dots, x_n\}$  ].

Definimos la función de correlación de las velocidades en donde la lista de velocidades se calcula en tiempos separados por lag, en diferencias finitas con  $\Delta=\text{deltaDif}$ , y donde el lag de la función de correlación ("lagDeCorrelacion") se da en unidades de lagCalculoVelocidad. Es decir, en unidades de números de frame, el lag de la función de correlación es igual a

lag\*lagDeCorrelacion ( $\rightarrow$  lag de correlación en frames)

```
In[30]:= autoCorrelacionV[particula_, lag_, deltaDif_, eje_, lagDeCorrelacion_] :=
CorrelationFunction[VLista[particula, lag, deltaDif, eje], lagDeCorrelacion]
```

```
In[31]:= autoCorrelacionV[508, 2, 2, "X", 2]
```

```
Out[31]= -0.444488
```

Con autoCorrelacionVLISTA generamos una lista de autocorrelaciones de la velocidad en la dirección eje para varios valores de lag de correlación (en unidades de lagCalculoVelocidad), yendo este valor de 0 a un valor máximo ("lagDeCorrelacionFinal") en incrementos iguales a lagDeCorrelacionIncremento

```
In[32]:= autoCorrelacionVLISTA[particula_, lagCalculoVelocidad_, deltaDif_,
  eje_, {lagDeCorrelacionFinal_, lagDeCorrelacionIncremento_}] :=
  Table[{lagDeCorrelacion, autoCorrelacionV[particula,
    lagCalculoVelocidad, deltaDif, eje, lagDeCorrelacion]}, {
    lagDeCorrelacion, 0, lagDeCorrelacionFinal, lagDeCorrelacionIncremento}]
```

```
In[157]:= autoCorrelacionVLISTA[160, 2, 1, "X", {30, 2}]
```

```
Out[157]= {{0, 1.}, {2, 0.71968}, {4, 0.538127}, {6, 0.558843}, {8, 0.509515}, {10, 0.260868}, {12, 0.112798},
{14, 0.151133}, {16, 0.0676291}, {18, -0.196686}, {20, -0.218038}, {22, -0.206666}, {24, -0.267179},
{26, -0.375932}, {28, -0.336486}, {30, -0.275143}}
```

Con `autoCorrelacionVLISTAFrames` generamos una lista de autocorrelaciones de la velocidad en dirección eje para varios valores de lag de correlación (en unidades de **frames**), yendo este valor de 0 a un valor máximo ("lagDeCorrelacionFinal") en incrementos iguales a `lagDeCorrelacionIncremento`

```
In[33]:= autoCorrelacionVLISTAFrames[particula_, lagCalculoVelocidad_, deltaDif_,
  eje_, {lagDeCorrelacionFinal_, lagDeCorrelacionIncremento_}] :=
  Table[{lagDeCorrelacion * lagCalculoVelocidad, autoCorrelacionV[
    particula, lagCalculoVelocidad, deltaDif, eje, lagDeCorrelacion]}, {
    lagDeCorrelacion, 0, lagDeCorrelacionFinal, lagDeCorrelacionIncremento}]
```

`autoCorrelacionVPromediada` proporciona una tabla con formato

**frame**, autocorrelación velocidad y average sobre todas las series con longitud mínima mayor o igual que "longitudMinimadelasSeries".

La autocorrelación se calcula para un `lagCalculoVelocidad` (en frames) dado . Ojo: `lagDeCorrelacionFinal` y `lagDeCorrelacionIncremento` se dan en unidades de `lagCalculoVelocidad`.

```
In[34]:= autoCorrelacionVPromediada[longitudMinimadelasSeries_, lagCalculoVelocidad_,
  deltaDif_, eje_, {lagDeCorrelacionFinal_, lagDeCorrelacionIncremento_}] :=
  autoCorrelacionVPromediada[longitudMinimadelasSeries, lagCalculoVelocidad,
  deltaDif, eje, {lagDeCorrelacionFinal, lagDeCorrelacionIncremento}] =
  Module[{listaParticulasSeleccionadas},
  listaParticulasSeleccionadas =
  listaParticulasconNumerodeDatosMayoroIgualque[
  longitudMinimadelasSeries];
  Mean[Table[autoCorrelacionVLISTAFrames[
  listaParticulasSeleccionadas[[m, 1]], lagCalculoVelocidad, deltaDif,
  eje, {lagDeCorrelacionFinal, lagDeCorrelacionIncremento}],
  {m, 1, Length[listaParticulasSeleccionadas]}]]]
```

```
In[35]:= etiquetaAvgV[longitudMinimadelasSeries_,
  lagCalculoVelocidad_, deltaDif_, eje_] := "Average (eje " <> eje <>
  " sobre series con long min=" <> ToString[longitudMinimadelasSeries] <>
  "\n lag V=" <> ToString[lagCalculoVelocidad] <> "con Δ=" <>
  ToString[deltaDif] <> " . Unidades abcisa: frames";
```

# Posición

## Análisis del tamaño de los saltos $\Delta x$ o $\Delta y$ y coeficiente de difusión

Análisis del tamaño de los saltos  $\Delta x$  o  $\Delta y$  para  $\text{deltan}=10$  y series con longitud mínima 20000

```
In[75]:=  $\text{deltan} = 10; \text{longituMin} = 20\ 000;$ 
 $\text{dataXCOMPLETA} = \text{serieTotaldeDeltaPosCOMPLETA}[\text{longituMin}, \text{deltan}, "X"];$ 
 $\text{dataYCOMPLETA} = \text{serieTotaldeDeltaPosCOMPLETA}[\text{longituMin}, \text{deltan}, "Y"];$ 
 $\{\{\text{Mean}[\text{dataXCOMPLETA}], \text{Variance}[\text{dataXCOMPLETA}], \text{Kurtosis}[\text{dataXCOMPLETA}],$ 
 $\text{Skewness}[\text{dataXCOMPLETA}]\}, \{\text{Mean}[\text{dataYCOMPLETA}], \text{Variance}[\text{dataYCOMPLETA}],$ 
 $\text{Kurtosis}[\text{dataYCOMPLETA}], \text{Skewness}[\text{dataYCOMPLETA}]\}\} // \text{TableForm}$ 

Out[75]/TableForm=

$$\begin{array}{cccc}
 -0.00130369 & 0.000851752 & 2.95161 & -0.092558 \\
 0.000180198 & 0.000897264 & 2.9306 & -0.0218374
 \end{array}$$


In[76]:=  $\text{deltan} = 10; \text{longituMin} = 20\ 000;$ 
 $\text{dataX} = \text{serieTotaldeDeltaPos}[\text{longituMin}, \text{deltan}, "X"];$ 
 $\text{dataY} = \text{serieTotaldeDeltaPos}[\text{longituMin}, \text{deltan}, "Y"];$ 
 $\{\{\text{Mean}[\text{dataX}], \text{Variance}[\text{dataX}], \text{Kurtosis}[\text{dataX}], \text{Skewness}[\text{dataX}]\},$ 
 $\{\text{Mean}[\text{dataY}], \text{Variance}[\text{dataY}], \text{Kurtosis}[\text{dataY}], \text{Skewness}[\text{dataY}]\}\} // \text{TableForm}$ 

Out[76]/TableForm=

$$\begin{array}{cccc}
 -0.00131038 & 0.000852012 & 2.95315 & -0.0903192 \\
 0.000177225 & 0.000898002 & 2.9295 & -0.0206965
 \end{array}$$


In[77]:=  $\{\text{distX} = \text{EstimatedDistribution}[\text{dataX}, \text{NormalDistribution}[\text{m}, \text{s}]],$ 
 $\text{distY} = \text{EstimatedDistribution}[\text{dataY}, \text{NormalDistribution}[\text{m}, \text{s}]]\}$ 

Out[77]=  $\{\text{NormalDistribution}[-0.00131038, 0.0291878],$ 
 $\text{NormalDistribution}[0.000177225, 0.0299652]\}$ 

In[78]:=  $\{\text{distXCOMPLETA} = \text{EstimatedDistribution}[\text{dataXCOMPLETA}, \text{NormalDistribution}[\text{m}, \text{s}]],$ 
 $\text{distYCOMPLETA} = \text{EstimatedDistribution}[\text{dataYCOMPLETA}, \text{NormalDistribution}[\text{m}, \text{s}]]\}$ 

Out[78]=  $\{\text{NormalDistribution}[-0.00130369, 0.0291847],$ 
 $\text{NormalDistribution}[0.000180198, 0.0299542]\}$ 

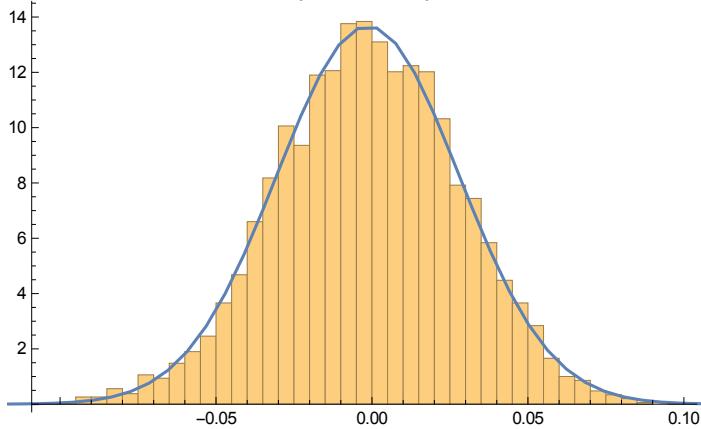
In[79]:=  $\{\text{FindDistributionParameters}[\text{dataX}, \text{NormalDistribution}[\text{m}, \text{s}]],$ 
 $\text{FindDistributionParameters}[\text{dataY}, \text{NormalDistribution}[\text{m}, \text{s}]]\}$ 

Out[79]=  $\{\{\text{m} \rightarrow -0.00131038, \text{s} \rightarrow 0.0291878\}, \{\text{m} \rightarrow 0.000177225, \text{s} \rightarrow 0.0299652\}\}$ 

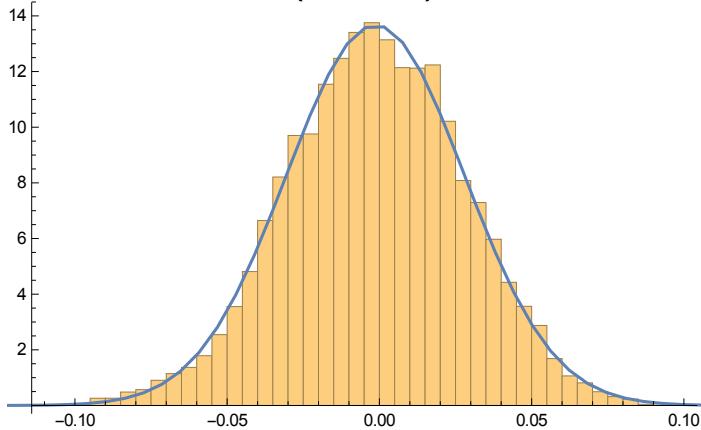
In[80]:=  $\{\text{FindDistributionParameters}[\text{dataXCOMPLETA}, \text{NormalDistribution}[\text{m}, \text{s}]],$ 
 $\text{FindDistributionParameters}[\text{dataYCOMPLETA}, \text{NormalDistribution}[\text{m}, \text{s}]]\}$ 

Out[80]=  $\{\{\text{m} \rightarrow -0.00130369, \text{s} \rightarrow 0.0291847\}, \{\text{m} \rightarrow 0.000180198, \text{s} \rightarrow 0.0299542\}\}$ 
```

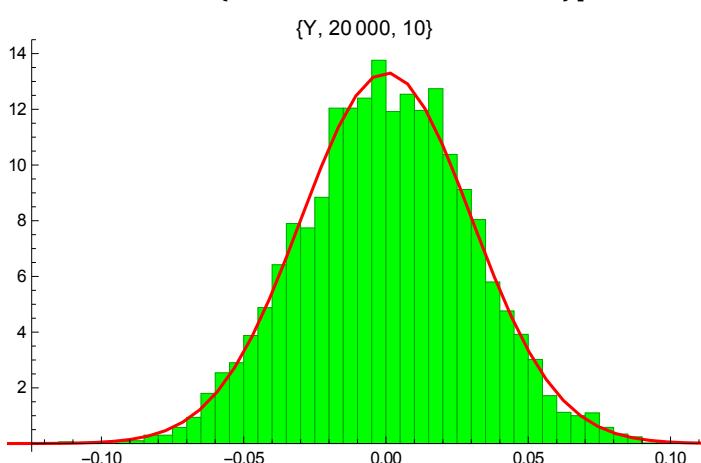
```
In[81]:= anchoCajas = 0.005;
k1 = Show[Histogram[dataX, {anchoCajas}, "ProbabilityDensity"],
Plot[PDF[distx, y], {y, -10, 10}], PlotLabel -> {"X", longituMin, deltan}]
{X, 20 000, 10}
```



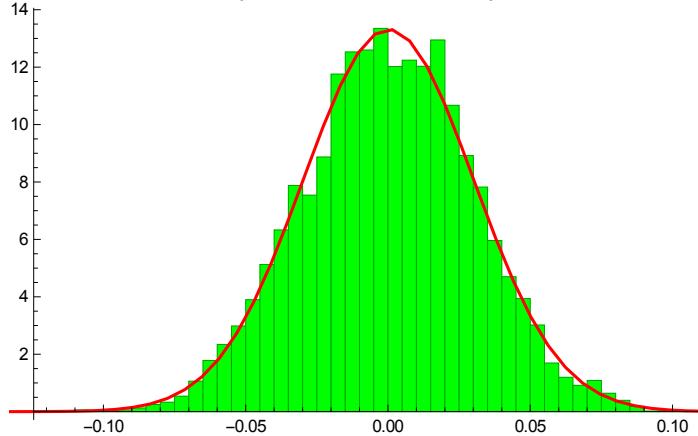
```
In[82]:= anchoCajas = 0.005;
k1 = Show[Histogram[dataXCOMPLETA, {anchoCajas}, "ProbabilityDensity"], Plot[
PDF[distXCOMPLETA, y], {y, -10, 10}], PlotLabel -> {"X", longituMin, deltan}]
{X, 20 000, 10}
```



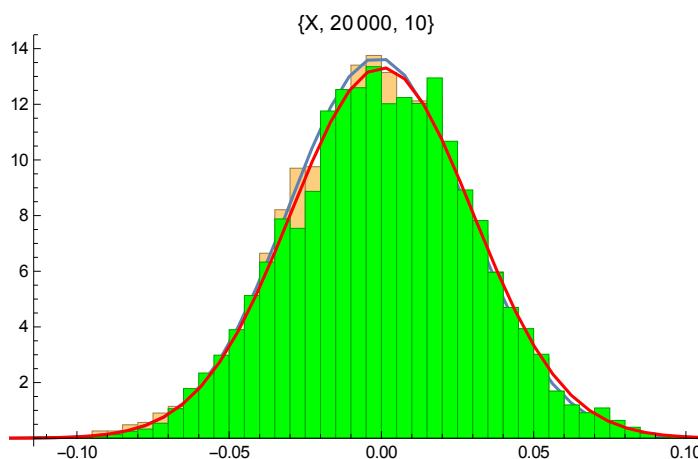
```
In[83]:= anchoCajas = 0.005; k2 =
Show[Histogram[dataY, {anchoCajas}, "ProbabilityDensity", ChartStyle -> Green],
Plot[PDF[disty, y], {y, -10, 10}, PlotStyle -> Red],
PlotLabel -> {"Y", longituMin, deltan}]
{Y, 20 000, 10}
```



```
In[84]:= anchoCajas = 0.005;
k2 = Show[Histogram[dataYCOMPLETA, {anchoCajas}, "ProbabilityDensity",
ChartStyle -> Green], Plot[PDF[distYCOMPLETA, y], {y, -10, 10}, PlotStyle -> Red],
PlotLabel -> {"Y COMPLETA", longituMin, deltan}]
{Y COMPLETA, 20000, 10}
```



```
In[85]:= Show[k1, k2]
```



A small p-value suggests that it is unlikely that the data came from dist.

```
In[86]:= HX = DistributionFitTest[dataX, distX, "HypothesisTestData"];
```

```
In[87]:= HX["TestDataTable", All]
```

	Statistic	P-Value
Anderson-Darling	1.03351	0.340138
Baringhaus-Henze	1.4728	0.0718248
Cramér-von Mises	0.12047	0.493523
Jarque-Bera ALM	14.4968	0.000987991
Kolmogorov-Smirnov	0.0101142	0.257962
Kuiper	0.0148944	0.176308
Mardia Combined	14.4968	0.000987991
Mardia Kurtosis	-0.956376	0.338882
Mardia Skewness	13.5959	0.000226678
Pearson $\chi^2$	96.912	0.0835222
Watson U <sup>2</sup>	0.090668	0.332232

```
In[88]:= HY = DistributionFitTest[dataY, distY, "HypothesisTestData"];
```

```
In[89]:= HYCOMPLETA =
DistributionFitTest[dataYCOMPLETA, distYCOMPLETA, "HypothesisTestData"];
```

In[90]:= HY["TestDataTable", All]

	Statistic	P- Value
Anderson- Darling	1.20129	0.26704
Baringhaus- Henze	2.67605	0.00291737
Cramér- von Mises	0.193376	0.280678
Jarque- Bera ALM	2.75332	0.256014
Kolmogorov- Smirnov	0.0122482	0.0995254
Kuiper	0.0211114	0.00409781
Mardia Combined	2.75332	0.256014
Mardia Kurtosis	-1.43911	0.150121
Mardia Skewness	0.71391	0.398149
Pearson $\chi^2$	122.608	0.00121641
Watson U <sup>2</sup>	0.187583	0.0493518

In[91]:= HYCOMPLETA["TestDataTable", All]

	Statistic	P- Value
Anderson- Darling	12.2333	$1.3477 \times 10^{-6}$
Baringhaus- Henze	31.6205	0.
Cramér- von Mises	1.98892	0.0000135349
Jarque- Bera ALM	27.9749	$8.42009 \times 10^{-7}$
Mardia Combined	27.9749	$8.42009 \times 10^{-7}$
Mardia Kurtosis	-4.47904	$7.49783 \times 10^{-6}$
Mardia Skewness	7.94432	0.00482384
Pearson $\chi^2$	1033.29	$9.86102 \times 10^{-113}$

In[92]:= {DistributionFitTest[dataX, distX, Automatic],  
DistributionFitTest[dataY, distY, Automatic]}

Out[92]= {0.493523, 0.280678}

In[93]:= {DistributionFitTest[dataXCOMPLETA, distXCOMPLETA, Automatic],  
DistributionFitTest[dataYCOMPLETA, distYCOMPLETA, Automatic]}

Out[93]= {0.000671427, 0.0000135349}

Análisis de tamaño de saltos para deltan=10 y series con longitud mínima 19000

Análisis de tamaño de saltos para deltan=10 y series con longitud mínima 10000

Coeficiente de difusión "microscópico" (o DifuSalto)

$$D_x = \text{Var}[\delta x] / (2 \delta t) \quad o \quad D_y = \text{Var}[\delta y] / (2 \delta t)$$

## TAMSD y coeficiente de difusión “macroscópico”

### Trayectorias x(t) o y(t) de varias partículas

walk1DY[nombreParticula\_, deltan\_] es una lista con las posiciones y de las partículas cada deltan frames

walk1DY[nombreParticula\_,deltan\_,{pasolni\_,pasoFin\_}] es una lista con las posiciones y de las partículas cada deltan frames como si la partícula hubiera empezado a andar en el pasolni y terminar en el pasoFin (pasos

en unidades de  $\delta t$ )

`caminata2D[nombreParticula_, $\delta t$ ] dibuja la trayectoria de la partícula  $\text{nombreParticula}$  en el plano 2D`

```
In[61]:= walk1D[nombreParticula_,  $\delta t$ , eje_] :=
  FoldList[Plus, 0, DeltaPosicionLista[nombreParticula,  $\delta t$ , eje]]
```

```
In[62]:= walk1D[nombreParticula_,  $\delta t$ , eje_, {pasoIni_, pasoFin_}] :=
  FoldList[Plus, 0,
    Take[DeltaPosicionLista[nombreParticula,  $\delta t$ , eje], {pasoIni, pasoFin}]]
```

```
In[84]:= caminata2D[nombreParticula_,  $\delta t$ ] :=
  ListPlot[Thread[{walk1D[nombreParticula,  $\delta t$ , "X"],
    walk1D[nombreParticula,  $\delta t$ , "Y"]}],
  PlotLabel → StringJoin["Caminata 2D (x,y) de partícula ",
    ToString[nombreParticula], " con pasos de tamaño ", ToString[ $\delta t$ ]]]
```

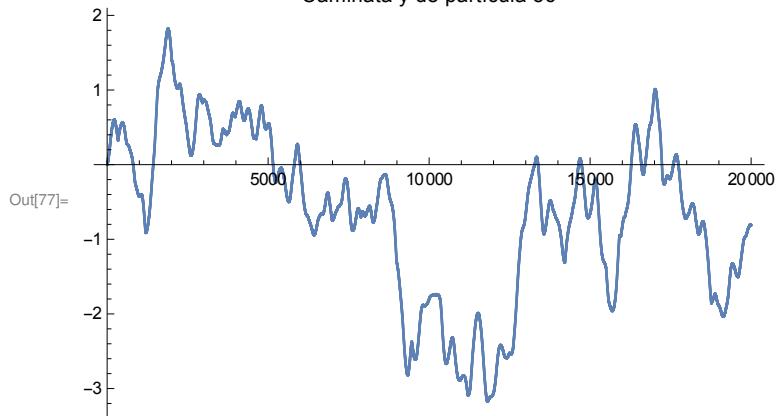
```
In[73]:= nombreParticula = 508; walk1D[nombreParticula, 1, "X"]
```

```
Out[73]= {0, -0.00401635, -0.00628903, -0.00970043, -0.0114915,
-0.0153243, -0.0180307, -0.0222488, -0.0251001, -0.0277756}
```

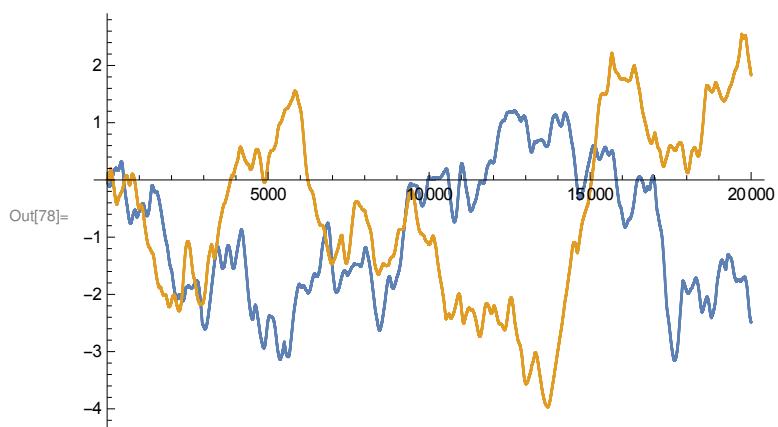
```
In[72]:= nombreParticula = 508; walk1D[nombreParticula, 1, "Y"]
```

```
Out[72]= {0, 0.00553049, 0.0166685, 0.0230718, 0.0327818,
0.0386737, 0.0493235, 0.0560045, 0.0662311, 0.0723233}
```

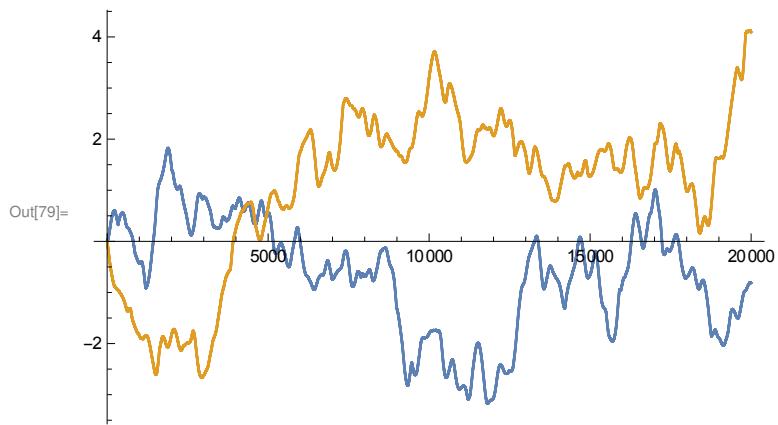
```
In[77]:= nombreParticula = 56; ListPlot[walk1D[nombreParticula, 1, "Y"],
  PlotLabel → StringJoin["Caminata y de partícula ", ToString[nombreParticula]]]
```



```
In[78]:= ejeV = "X"; ListPlot[{walk1D[56, 1, ejeV], walk1D[2, 1, ejeV]}]
```

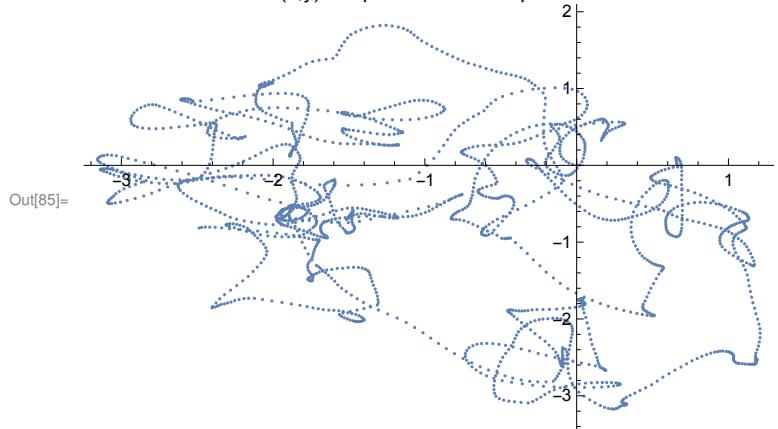


```
In[79]:= ejeV = "Y"; ListPlot[{walk1D[56, 1, ejeV], walk1D[2, 1, ejeV]}]
```



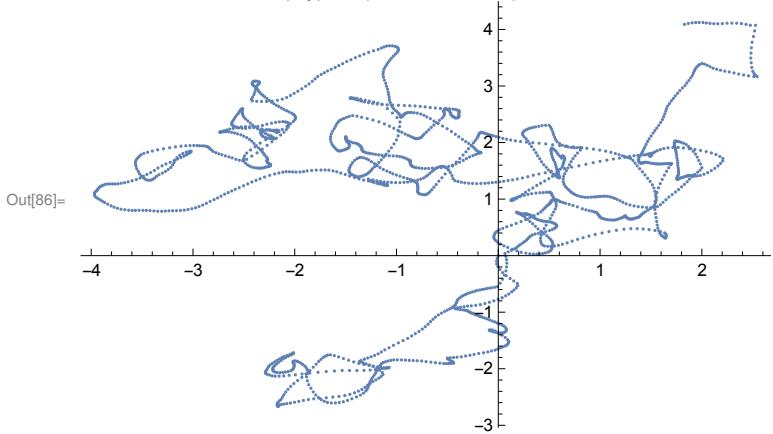
```
In[85]:= caminata2D[56, 10]
```

Caminata 2D (x,y) de partícula 56 con pasos de tamaño 10



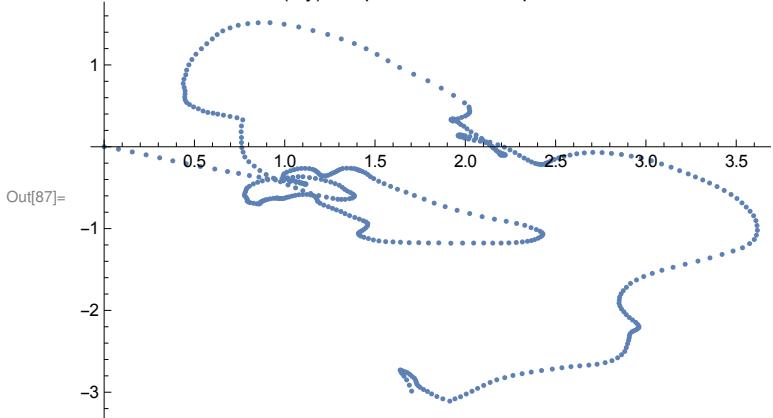
```
In[86]:= caminata2D[2, 10]
```

Caminata 2D (x,y) de partícula 2 con pasos de tamaño 10



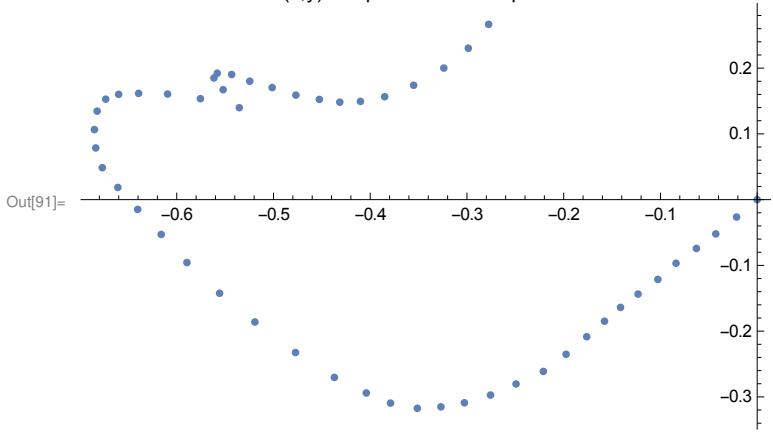
```
In[87]:= caminata2D[11, 10]
```

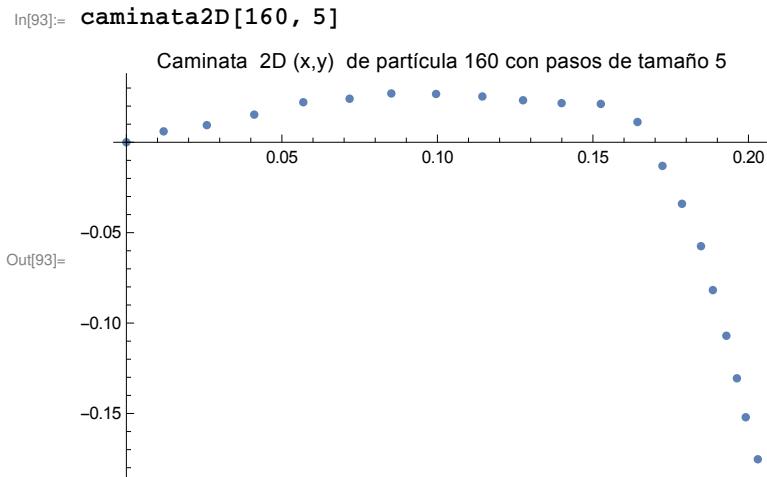
Caminata 2D (x,y) de partícula 11 con pasos de tamaño 10



```
In[91]:= caminata2D[0, 10]
```

Caminata 2D (x,y) de partícula 0 con pasos de tamaño 10





## Velocidad

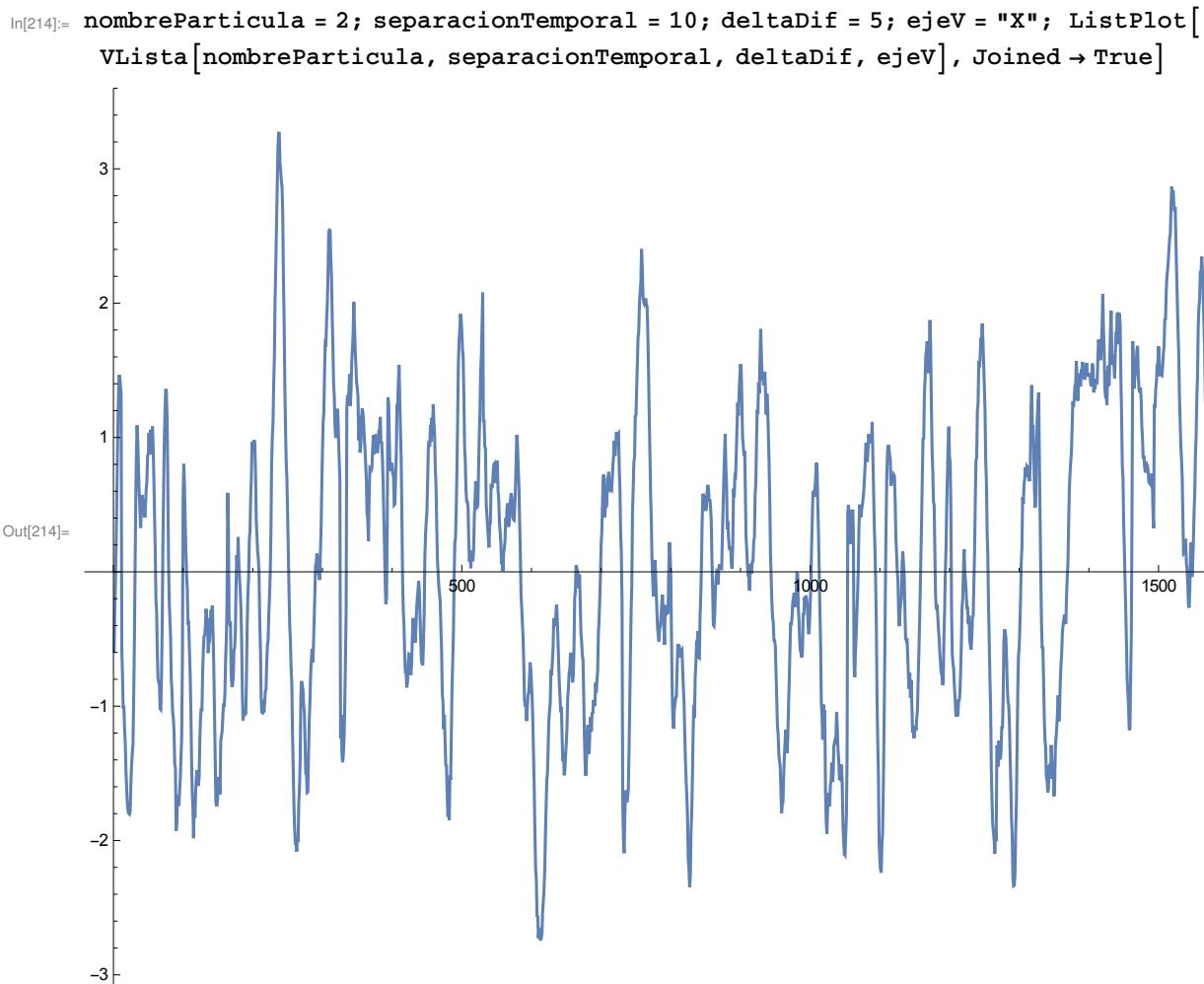
Velocidad en la dirección y en unidades de diámetros de ppp por segundo.

Se usa fórmula diferencias central de dos puntos con distancia derecha-izquierda igual a +- lag

Ejemplos

```
In[186]:= nombreParticula = 505; separacionTemporal = 2; deltaDif = 1; ejeV = "X";
VLista[nombreParticula, separacionTemporal, deltaDif, ejeV]
Out[186]= {0.835618, 0.863432, 0.59742, 1.52699, 0.914652, 1.01719, 1.42424, 1.26085,
-1.45047, -0.649699, -0.867041, -0.620655, -0.421581, -0.455914,
-0.462283, -0.439052, -0.684437, -0.332018, -0.37913, -0.142717,
-0.358327, -0.319829, -0.148107, -0.200073, -0.0522342, -0.142853,
0.133153, -0.170413, 0.135492, -0.0274498, -0.183185, 0.195234}

In[195]:= nombreParticula = 505; separacionTemporal = 2; deltaDif = 1;
ejeV = "X"; frameInicial = 1; frameFinal = 16; VLista[nombreParticula,
separacionTemporal, deltaDif, ejeV, {frameInicial, frameFinal}]
Out[195]= {0.863432, 0.59742, 1.52699, 0.914652, 1.01719, 1.42424}
```



En las figuras siguientes se muestra la velocidad frente al tiempo y se observa:

- 1) que la velocidad no es constante con cambios bruscos, lo que se esperaría si las partículas se comportaran como ppp que se mueven libremente entre choques
- 2) Las velocidades cambian de frame a frame, lo que indica que hay aceleraciones.
- 3) las velocidades cambian de una forma relativamente lineal entre intervalos que se separan por cambios bruscos de la velocidad.
  - 3a) Hipótesis a: Los cambios bruscos pueden deberse \*principalmente\* a choques entre las ppp
  - 3b) Hipótesis b: Los cambios bruscos pueden deberse a choques entre las ppp \*y\* fuerzas hidrodinámicas turbulentas.
  - 3b) Entre los choques (o cambios bruscos turbulentos) la aceleración es relativamente constante y por eso la velocidad en escalas de (grosso modo) unos 100 frames=0.25s cambia linealmente.

Calculo de la varianza de la componente x o y de la velocidad. Depende notablemente del valor del lag usado en su cálculo.

Histograma de velocidades de partículas individuales

Histograma (UNIFICADO) de velocidades de partículas con serie de datos con una longitud mínima

Histograma (UNIFICADO) de velocidades de partículas con serie de datos con una longitud mínima

Histograma (UNIFICADO) y ANÁLISIS de velocidades de partículas con serie de datos con una longitud mínima

---

## Autocorrelación de velocidades

*La autocorrelación se reduce en un factor 1/e después de unos 100 frames, i.e., 1/4 segundos, aproximadamente. Tras 20\*10 frames=1/2 segundo cae hasta casi cero y se mantiene después “oscilando-fluctuando” en torno a cero.*

```
RefLink[ CorrelationFunction , paclet : ref / CorrelationFunction ] [  

{ $x_1, \dots, x_n$ },  $h$ ] is equivalent to  $\sum_{i=1}^{n-h} (x_i - \hat{\mu}) (x_{i+h} - \hat{\mu}) / \sum_{i=1}^n (x_i - \hat{\mu})^2$  with  $\hat{\mu} =$   

RefLink[ Mean , paclet : ref / Mean ] [{ $x_1, \dots, x_n$ }].
```

Autocorrelación de velocidades Vx o Vy de partículas individuales

Autocorrelación de velocidades Vx o Vy **promediada** sobre partículas/series distintas